☣ Do not hand in paper copies. Instead, E-mail your solution to koppel@ece.lsu.edu. The preferred format is a PDF file.

Problem 1: *Note: The following problem was assigned in each of the last five years, and its solution is available. DO NOT look at the solution unless you are lost and can't get help elsewhere. Even in that case just glimpse.* Appearing below are **incorrect** executions on the illustrated implementation. For each one explain why it is wrong and show the correct execution.
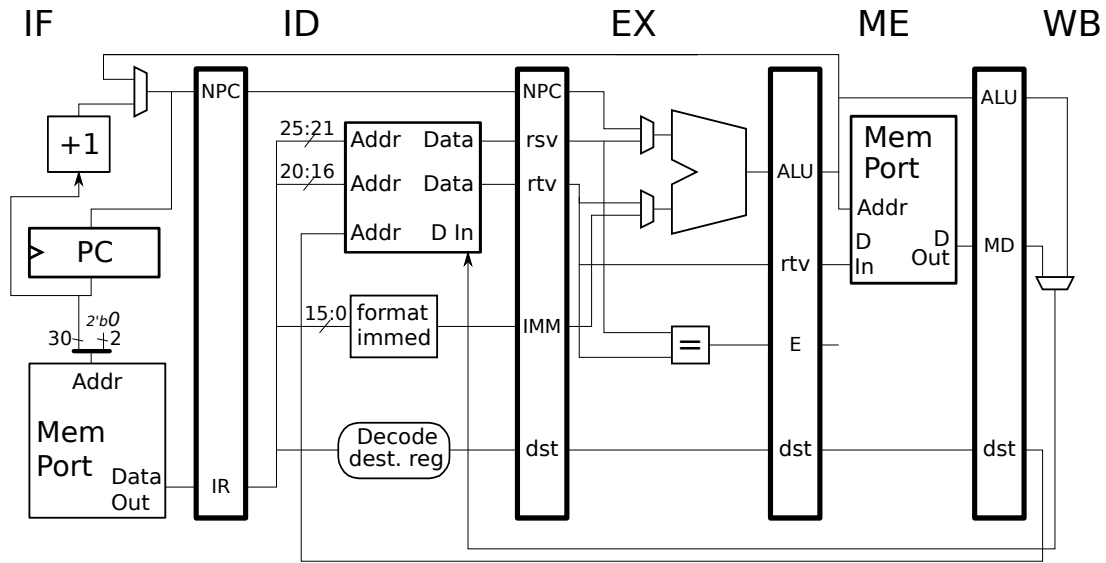
(*a*) Explain error and show correct execution.

```
# Cycle          0  1  2  3  4  5  6  7
lw r2, 0(r4)     IF ID EX ME WB
add r1, r2, r7      IF ID EX ME WB
```

(*b*) Explain error and show correct execution.

```
# Cycle          0  1  2  3  4  5  6  7
add r1, r2, r3   IF ID EX ME WB
lw r1, 0(r4)        IF ID -> EX ME WB
```
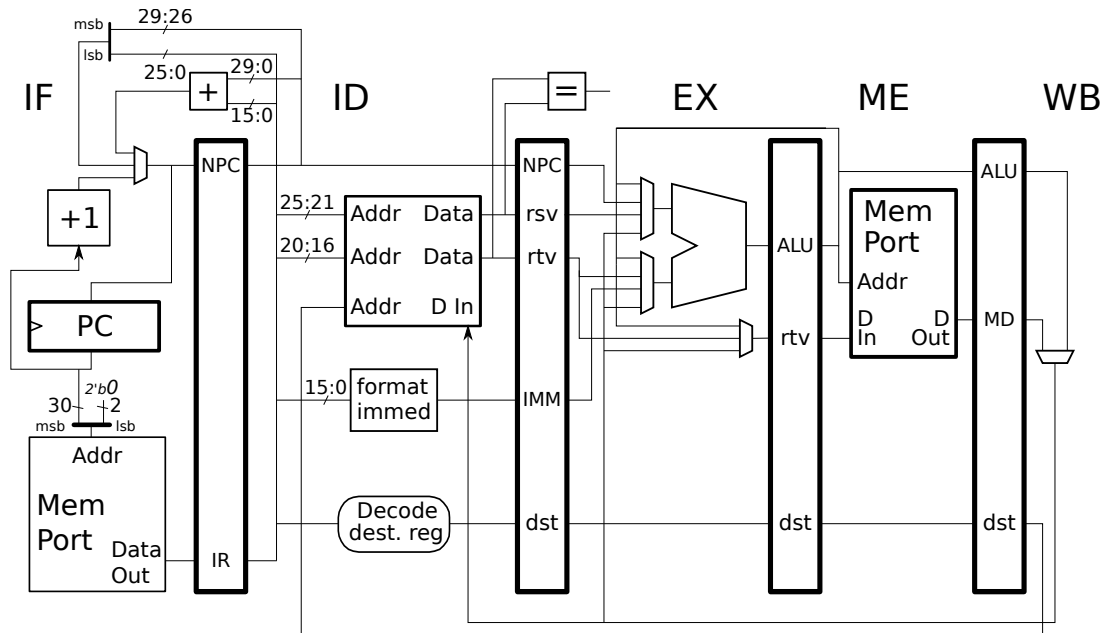
(*c*) Explain error and show correct execution.

```
# Cycle          0  1  2  3  4  5  6  7
add r1, r2, r3   IF ID EX ME WB
sw r1, 0(r4)        IF ID -> EX ME WB
```

(*d*) Explain error and show correct execution.

```
# Cycle          0  1  2  3  4  5  6  7
add r1, r2, r3   IF ID EX ME WB
xor r4, r1, r5      IF ----> ID EX ME WB
```

Problem 2: Appearing below are **incorrect** executions on the illustrated implementation. Notice that this implementation is different than the one from the previous problem. For each execution explain why it is wrong and show the correct execution.

(*a*) Explain error and show correct execution.

```
# Cycle          0  1  2  3  4  5  6  7
lw r2, 0(r4)     IF ID EX ME WB
add r1, r2, r7      IF ID EX ME WB
```

(*b*) Explain error and show correct execution.

```
# Cycle          0  1  2  3  4  5  6  7
add r1, r2, r3   IF ID EX ME WB
lw r1, 0(r4)        IF ID -> EX ME WB
```

(*c*) Explain error and show correct execution.

```
# Cycle          0  1  2  3  4  5  6  7
add r1, r2, r3   IF ID EX ME WB
sw r1, 0(r4)        IF ID -> EX ME WB
```

(*d*) Explain error and show correct execution. Note that this execution differs from the one from the previous problem.

```
# Cycle          0  1  2  3  4  5  6  7
add r1, r2, r3   IF ID EX ME WB
xor r4, r1, r5      IF ID ----> EX ME WB
```

Problem 3: Appearing below is the labeled MIPS implementation from 2018 Midterm Exam Problem 2(b), and as in that problem each mux in the implementation below is labeled with a circled letter, and mux inputs are numbered. Some wires are colored to make them easier to follow. Write code sequences that use the mux inputs as requested below. Some code sequences may consist of a single instruction.
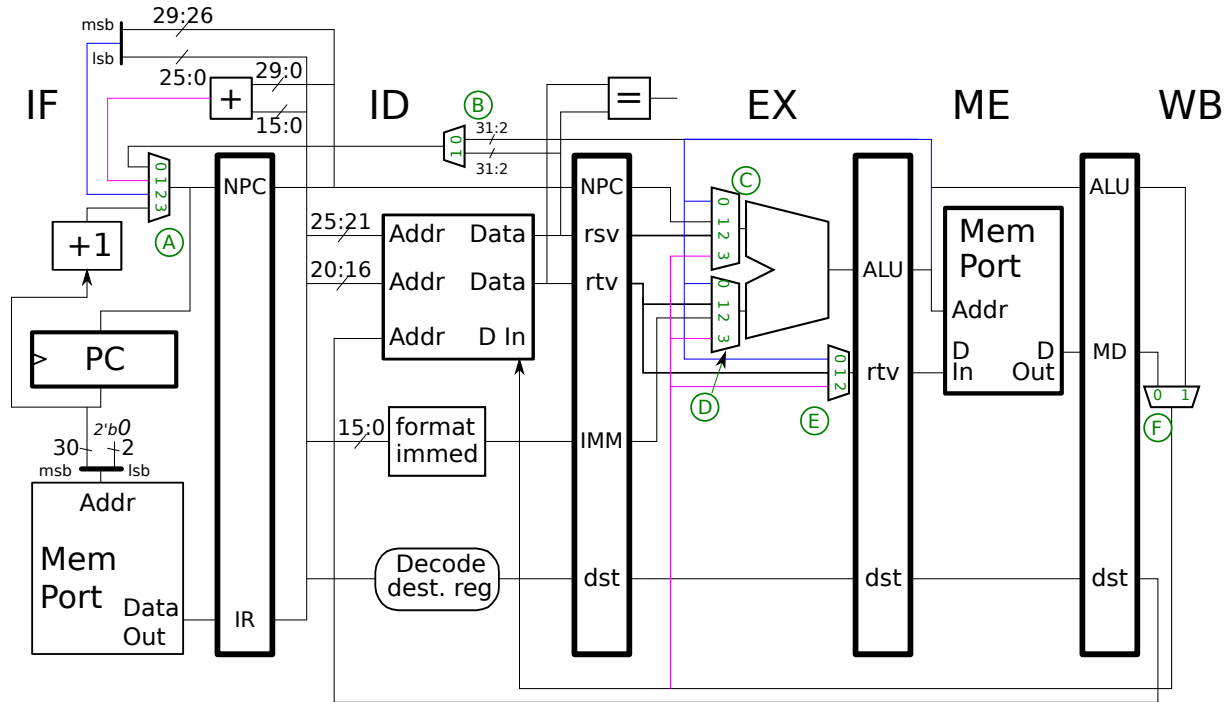
(a) Use F0.

(b) Use F0 and C3 at the same time. The code **should not** suffer a stall.

(c) Explain why its impossible to use E0 and D0 at the same time.

Problem 4: *This problem appeared as Problem 2c on the 2020 final exam.* Appearing below is our bypassed, pipelined implementation followed by a code fragment.

It might be helpful to look at Spring 2019 Midterm Exam Problem 4a. That problems asks for the execution of a loop and for a performance measure based upon how fast that loop executes.

(*a*) Show the execution of the code below on the illustrated implementation up to the point where the first instruction, `addi r2,r2,16`, reaches `WB` in the second iteration.

(*b*) Based on your execution determine how many cycles it will take to complete n iterations of the loop.
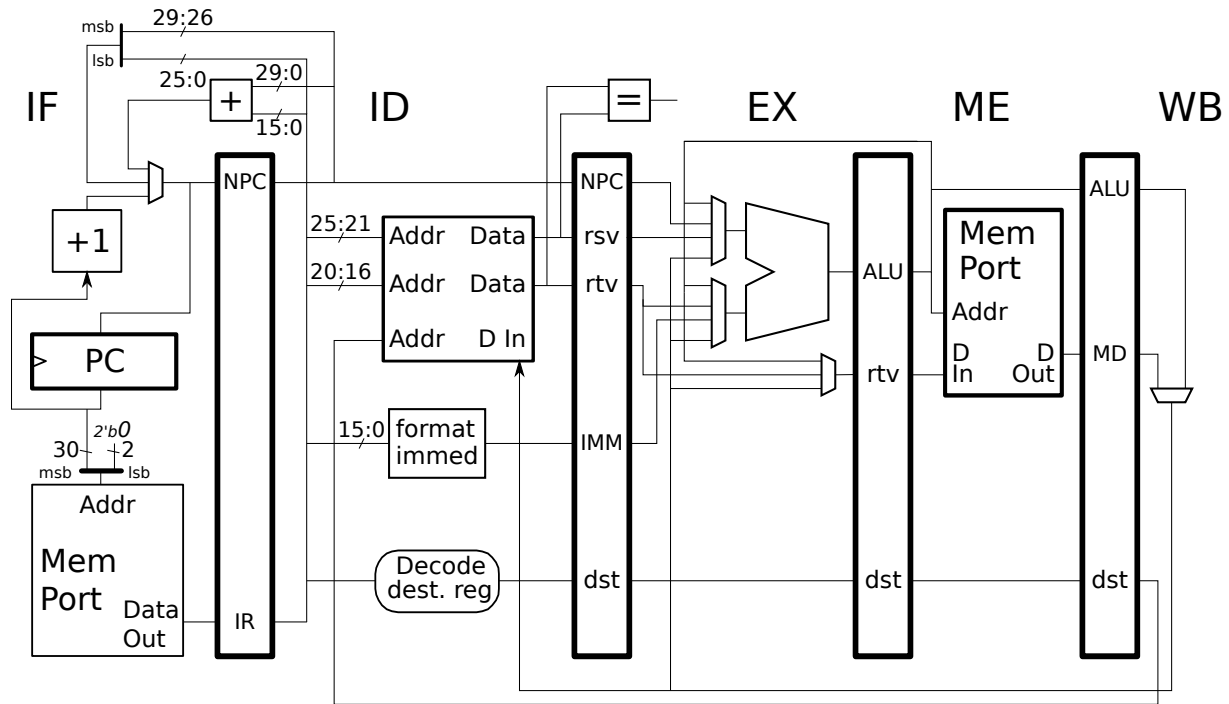

```
LOOP:
 addi r2, r2, 16

 lw r1, 8(r2)

 sw r1, 12(r3)

 bne r3, r4, LOOP

 addi r3, r3, 32

 sub r10, r3, r2
```